



Forshaw M, Solaiman E, McGee O, Robinson P, Emerson R. [Meeting Graduate Employability Needs through Open-source Collaboration with Industry](#). *In: ACM SIGCSE '16*. 2016, Memphis, TN, USA: ACM.

Copyright:

© Forshaw, M et al. | ACM. 2016. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in SIGCSE '16 Proceedings of the 47th ACM Technical Symposium on Computing Science Education.

DOI link to article:

<http://dx.doi.org/10.1145/2839509.2844649>

Date deposited:

11/07/2017

Meeting Graduate Employability Needs through Open-source Collaboration with Industry

Matthew Forshaw, Ellis Solaiman,
Oonagh McGee, Hugo Firth
School of Computing Science
Newcastle University, United Kingdom
{matthew.forshaw,ellis.solaiman,
oonagh.mcgee,h.firth}@ncl.ac.uk

Paul Robinson, Ryan Emerson
Red Hat Inc
Newcastle upon Tyne, United Kingdom
{paul.robinson,remerson}@redhat.com

ABSTRACT

This paper describes the development and delivery of a course, in close collaboration with industry, over a ten-year period. We describe the details of this collaboration, which aims to equip students with collaborative software development experience, incorporating open source, *Enterprise Middleware* technologies and industry best practices. In presenting our case study we share valuable insights, into the challenges and opportunities, for all parties involved. We present the reflections of all stakeholders, including; the course leader, the industrial partner, lab demonstrators, and students. We also present the results of our investigations to track the employment of our graduates over the last five years, identifying that all students who responded are employed within relevant positions or undertaking further study. Based on our successful delivery of the course, in this paper we make all tools and teaching materials available to the open source community, for delivery at other institutions. We emphasise the importance of the continued involvement of industry partners to inform pedagogical practices within Computer Science.

Keywords

industrial collaboration; outreach; employability; middleware; pedagogy; open source

1. INTRODUCTION

The software industry is an ever-changing and competitive environment, and expectations for graduate software engineers entering industry are high. Highly skilled graduates are key resources to drive continuous innovation and development of new computing applications and technologies. This provides a unique opportunity for CS educators to continually reflect and redesign the curriculum in order to deliver learning methods and resources capable of providing students with skills that are relevant to industry.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCSE '16, March 02 - 05, 2016, Memphis, TN, USA

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-3685-7/16/03...\$15.00

DOI: <http://dx.doi.org/10.1145/2839509.2844649>

Graduates are expected to possess a blend of cognitive, applied and interpersonal skills [14, 17, 8]. This presents a pedagogical challenge; *how do we balance the teaching of theoretical frameworks whilst empowering students with a holistic set of skills to succeed in this environment?*

Academia acknowledges the need to enhance programmes to equip students with skills for employment, yet there exists a disparity between industry expectations and existing curriculum frameworks [15]. This can be addressed by increased involvement of industry partners in the design and development of computing science programmes; important factors determining the success of such initiatives are highlighted in the literature, including the proximity of academic institution to industry partner [6], and the delivery of material in a real-world context [11].

In this paper we present a unique case study of over ten years of collaboration, between the School of Computing Science, Newcastle University and industrial partner, Red Hat [1], the world's largest provider of open source solutions.

We specifically describe the collaboration between Newcastle University and Red Hat on the design and delivery of the *Enterprise Middleware* course within our Advanced MSc programmes. The uniqueness of this study, compared to other capstone courses, stems from a long standing relationship, and from the close proximity of the school to the Red Hat Cloud Research Centre, created in 2010 at Newcastle University¹, co-located on Newcastle University's new Science Central campus. This initiative drives investment in research on enterprise technologies such as cloud, virtualisation, and middleware technologies.

The remainder of this paper is organised as follows. Relevant related work is presented in Section 2, and we describe our teaching initiative in Section 3. In Section 4 we highlight key challenges of running a course in collaboration with industry, identifying potential solutions to these. Section 5 presents student, academic, and industrial perspectives on the benefits of delivering a collaborative course. We explore student satisfaction through University-administered surveys and volunteered testimonials from recent graduates in Section 6. Section 7 presents student outcomes and employability data, and provides a recommendation for further longitudinal study. Section 8 presents our tools and teaching materials, available to the open source community for delivery at other institutions.

¹<http://www.redhat.com/en/about/press-releases/2963>

2. RELATED WORK

A number of initiatives which seek to bring industrial relevance to computer science courses, are presented in the literature. The mode and extent of industry involvement in the design and delivery of these courses varies significantly.

A great body of work relates to academic initiatives seeking to emulate industry through cross-site project work. Rusu *et al* [15] provide a survey of previous efforts, and describe their initiative, combining theoretical software engineering content with an industry-informed collaborative working in paired teams across multiple universities.

Collaborations have also been initiated by industry, e.g. Chandran *et al* [3] presents work by Intel, to promote the field of electronics packaging in Malaysian university programmes, leading to the employment of over thirty students.

Further initiatives seek to expose students to industrial practices outside the structure of a degree programme, e.g. Larsen *et al* [7] present a multi-disciplinary summer school delivered in collaboration with Bang and Olufsen, Denmark, and emphasise the importance of students developing soft competencies in multi-disciplinary international teams.

Other studies obtain input to curriculum design through industry advisory boards, and guest speakers [18]. Tenenbergs [16] provides an alternative argument and presents the *Industry Fellows* model, emphasising the importance of industry involvement in the classroom, and describes a course featuring an industry partner from Google as a co-lecturer.

Our initiative demonstrates novelty compared to those in the literature in the following aspects; *a)* closeness of collaboration and proximity between academia and the industry partner, *b)* we present insights gained through a decade-long initiative, *c)* we document issues around enhancing future iterations of the course, *d)* we make our tools and teaching materials available open source for use by other educators.

3. COURSE DESCRIPTION

The initiative we describe in this paper forms the practical component of an ‘Enterprise Middleware’ course (or ‘module’) delivered to our Advanced MSc students, including UK, EU and International students. The course runs for four weeks, split evenly between two weeks (16 hours) of practical work, and 2 weeks (16 hours) of lectures. Assessment for the course comprises a 90-minute closed-book, summative written examination at the end of Semester 1, and a programming assignment and accompanying report worth 30% of the course mark. While the written examination focuses on theoretical aspects of distributed computing, the coursework assignment is a highly practical application of these concepts and principles, which also seeks to incorporate industry best practices. This close relationship between lecture/examination and practical components embodies the principles of constructive alignment and constructivist learning theory [2], i.e. *active learning* and *collaborative learning*.

This coursework covers the development of an enterprise application using Java Platform Enterprise Edition (Java EE) technologies deployed to a Platform as a Service (PaaS) cloud. Students are introduced to modern production-grade enterprise middleware technologies and their application to build a multi-tier distributed ‘holiday booking’ application. Hawthorne *et al* [5] acknowledge the benefits of teaching such integration platforms to reinforce students’ knowledge of architectural styles, patterns, techniques and principles.

We aim to reflect industry, where developers commonly work on existing codebases; we provide students with a significant ‘base project’ which they should extend to provide additional functionality, described in a detailed specification. This is often a student’s first experience working with someone else’s code, especially at this scale. We assist the students throughout the challenge of familiarising themselves with the base code, by providing detailed guidance materials and comprehensive source code documentation.

The assignment begins with individualised self-guided work, supported by group discussion, where students develop a cloud-based booking service (e.g. ‘Taxi’, ‘Flight’, ‘Hotel’). In a final exercise, students are required to work collaboratively, to integrate their solutions with those of two colleagues in order to develop an integrated holiday booking system. Informal formative feedback is offered, through demonstrator support, during 16 hours of supervised practical labs. Assessment focuses on the tangible artefacts produced through collaboration, acknowledging complexities in assessing the process [13]. Students are expected to produce a short written report, providing technical documentation of their solution, demonstrate understanding of core concepts, and to reflect on their experiences and personal learning.

In addition to the summative assessment of submitted source code and report, students undertake a 15-minute demonstration/viva held two days before the submission deadline. Vivas are an informal process where students describe their technical solution and experiences on the course, are asked open questions related to core concepts, and receive formative feedback prior to submission.

The purpose of the assignment goes beyond a means of assessment, but seeks to empower students with skills and real-world experience developing a large software artefact, resources not available through other courses. Students’ solutions are intentionally ‘full stack’, comprising front- and backend technologies, such that they may form a compelling portfolio piece when seeking employment. This offers a unique opportunity for the students to demonstrate their abilities and use this for their continued career progression.

4. CHALLENGES AND OPPORTUNITIES

A number of challenges exist when delivering courses with industry which exhibit significant technical complexity and collaborative elements. In this section we discuss our course, challenges encountered, and document the interventions we put in place to ameliorate their effects.

4.1 Development of course materials

Generalisability: A common issue, where industry collaboration is concerned, surrounds the generalisability of industry-informed taught content. Care must be taken to ensure that external input is relevant on a wider scale, and not specific to the industrial partner involved. This is particularly important when considering international cohorts, whose employment destinations may span global job markets, each exhibiting cultural variation.

We are confident students’ expertise gained on the course may be easily generalised, due to; *a)* we focus on Java EE and cloud technologies which see widespread industrial use, *b)* Red Hat tools are open source and are reference implementations of Java standards, and *c)* ongoing efforts to ensure clear links between the practical elements of the course and theoretical concepts. This objective is shared by our indus-

try partner; “We have been very careful to ensure that Red Hat technologies are simply used as a vehicle to learning generally applicable skills and technologies.”

Scalability: The issue of scalability is often underestimated when formulating delivery and assessment methods, and selecting the level of difficulty for a course. Techniques appropriate for a particular size of cohort may be intractable if student numbers rise. The need for lightweight assessment mechanisms is felt particularly when collaborating with industry partners, whose availability is often limited.

As the complexity of coursework rises, so too does its assessment. For example; in the context of distributed computing, significant environment setup complicates being able to execute student code in a time efficient manner. This issue is exacerbated as the number of students rises.

In response to a significant rise in student intake in recent years, we have successfully employed the ‘viva’ assessment activity, as described in Section 3. The vivas were well received by students, and demonstrators found them extremely effective in assessing understanding, identifying cases of collusion, and ensuring the fairness of marking. Adding the viva assessment means we have been able to retain complexity while simplifying marking.

Accessibility: It is trivial to formulate industry-inspired curricula focused towards more experienced and competent students. A much greater challenge is ensuring that the material remains accessible to less capable students while retaining the intended learning outcomes. This represents a trade-off between realism and difficulty; making the course too difficult would alienate some students, while oversimplifying aspects of the course may overly protect students from real-world issues.

We believe our initiative should represent a positive experience for all students, allowing them to gain confidence in their real-world software development experience. Hence, we place a strong emphasis on providing students with *ipsative feedback*, acknowledging the personal learning experience of each student in the assessment process [9]. We also provide students the opportunity to reflect on their experiences through the viva and written report assessment components.

Providing an environment where the teaching methods and assessment are accessible to all, poses a challenge. We seek to combat this through these blended assessment methods, and collaborative nature of the course.

4.2 Reconciling academic and industry needs

In designing a course with industry involvement, it is essential to reconcile the two, sometimes conflicting, viewpoints of academia and industry. Academics need to demonstrate the application of theoretical concepts using current technologies, and professionals need to ensure they focus less on the minutiae of a particular technology, but the broader learning outcomes.

Key issues surround managing the expectations of both parties, with regards to establishing an appropriate level of difficulty of the course, and ensuring the assignment accurately reflects industrial practice, whilst enhancing and adhering to university regulations.

In the context of this course, our industrial partner possesses Masters-level and PhD degrees from our academic institution, so understands the academic context we operate within, and what is required from our students. Therefore, in the design of our ‘Enterprise Middleware’ course, we in-

volve industry partners closely in the design of the marking scheme and validation of the assessment. Our industry partner also engages at the marking stage, co-marking a subset of the students’ work to ensure fairness of marking and contributing to written summative feedback. This moderation affords a unique opportunity for the integration (in the mark scheme) of industry perspectives with academic learning outcomes. In doing so, we provide students with feedback and feed forward specifically informed by industry input. We advocate academic partners actively engaging with industry prior to, during, and following the course.

4.3 Maintainability

Operating within the ever-changing environment of industry best practice and tooling, necessitates keeping course content up-to-date. This requires significant development work by university staff, in consultation with Red Hat to ensure the *base project* provided to students supports desired learning outcomes. By selecting a *lead demonstrator* to liaise with the industrial partner and perform required changes, three months prior to the beginning of the course, we are able to ensure the smooth delivery of the course.

In addition to the effort required to keep the course in line with current industry best practices, significant up-front development effort has been required each year to ensure the compatibility of the coursework solution, as software versions, cloud platforms and local infrastructure changes.

We have recently ameliorated these issues by contributing the *base project* to the open-source community as a JBoss Developer “Quickstart”, a “*use-case code examples tested with the latest stable product releases*”². The maintenance of the project will now be carried out by the industry partner, ensuring compatibility with subsequent software releases. We retain responsibility for updating additional documentation, the coursework specification and model solutions.

Great care is taken to ensure the course could endure a change in personnel. For our demonstrators we encourage an apprenticeship model, whereby the current lead demonstrator trains new demonstrators to take their place in subsequent years. This is important when drawing from a pool of PhD students, who change every three or four years.

The issue of staff turnover is a risk common to all courses which are delivered with industry participation. There has been significant investment by Red Hat in the development of the course over a ten-year period, alongside two changes in academic course leader. In light of this, we are confident of ongoing industry support if staff changeover were to occur.

5. STAKEHOLDER REFLECTION

In order for the continued success and effective delivery of the course it is necessary to actively engage with key stakeholders. These include the demonstrators, course leader, industry partners and students.

5.1 Demonstrators

The delivery of practical labs on the course relies heavily on the support of demonstrators (graduate teaching assistants), comprising PhD students from the School of Computing Science. Whilst demonstrators provide a valuable contribution to courses across the department, their contribution to this course is integral to its operation.

²<http://www.jboss.org/developer-materials/>

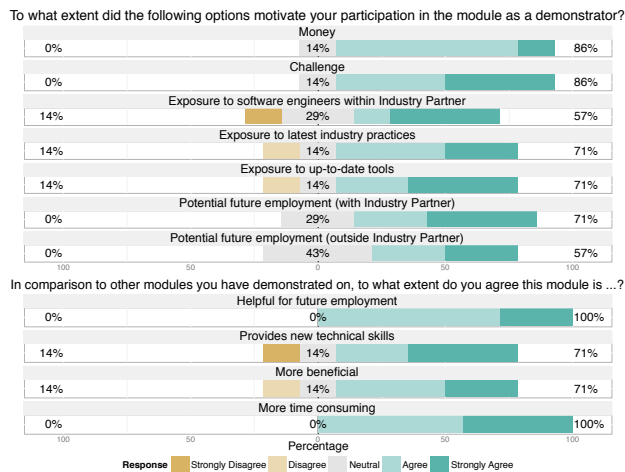


Figure 1: Demonstrator survey results

This course differs in that it requires significant advanced knowledge and substantial preparatory work to deliver the course. Demonstrators commonly acknowledge that demonstrating on this course is more time consuming and challenging than other courses. Despite these potential barriers to demonstrator recruitment, each year we have been successful in recruiting the required demonstrators to assist on the course, with all demonstrators participating in multiple consecutive years throughout their PhDs.

In order to better understand demonstrators's motivations for demonstrating on the course, and in doing so secure sustainable demonstrator provision, we have surveyed every demonstrator on the course between 2010 and 2015, with results presented in Figure 1. The width of the figure represents 200% of the sample size, with percentage values showing the proportion of negative, neutral and positive responses respectively. Each row is horizontally aligned so that neutral responses are centred, in order to more easily observe overall sentiment.

Although "financial incentives" were cited as being a significant motivator for demonstrating on this course, students were not financially remunerated proportionally for the additional workload required in comparison to demonstrating on other courses within the department. Hence we believe further factors motivate students. We see demonstrators value exposure to latest industry practices and tools, and consider their participation to be beneficial to career prospects, with the industry partner and also generally.

We supplement our survey results with volunteered testimonials from demonstrating staff involved in the course between 2010 and 2015; one demonstrator who was previously a student on the course, and now works at Red Hat, comments; *"I felt that the module was the most beneficial to me during my masters. I wanted to contribute to help new students in the hope they would gain as much as me."* The same demonstrator goes on to cite *"contributions to the open-source community"* as an additional factor motivating his participation on the course.

Finally, demonstrators reported that their participation on the course enhanced their own skills, e.g. *"helps PhD students keep up to date with industry trends outside of their research area"*. Further empirical evidence supports that the content is useful to our PhD research students, with a

number of doctoral students requesting to attend each year.

Our survey results and empirical evidence suggests our course benefits not only the students enrolled on the course, but also empowers our demonstrators, enhancing them as lifelong learners through continued skill development.

5.2 Lead Demonstrator

One *lead demonstrator* (also a graduate teaching assistant) each year is responsible for *a)* co-ordinating and administering demonstrator provision, *b)* liaising with the industry partner on developing the course materials, *c)* allocation and moderation of marking.

"The experience of acting as lead demonstrator on this module posed significant additional workload, but afforded me the opportunity to enhance my knowledge of industry-specific tools and provided me with improved technical skills."

"The module has exposed me to the tools and techniques required for curriculum development and provided me the opportunity to actively contribute to the development and enhancement of the assessment framework. Within the development of the assessment a further challenge arose, that of the individual student learning journey. As a consequence, additional fluidity was introduced into the marking scheme."

"This combined experience proved invaluable when applying for a lecturing position on completion of my PhD."

5.3 Course Leader

The course leader is responsible for delivering lectures, and for the coordination of course activities (e.g. deciding course content, methods of delivery, and assessment), and of ensuring that the course aim and objectives are met. Here the course leader speaks of the benefits of industry collaboration in relation to the delivery of the course, and the wider impact of this engagement on research supervision.

"Ongoing collaboration with Red Hat industry leaders in open-source middleware technologies, allows for the curriculum to be continually informed by the current needs of industry. This enables us as educators to energise teaching materials to reflect latest middleware technologies and development methodologies."

"This collaboration has further exposed me to current research challenges, enabling me to provide research supervision at Masters and PhD levels which exhibit greater industrial relevance and impact."

5.4 Industry

The involvement of industry is driven by two key objectives; seeding technologies, and talent acquisition. Within this section we consider these and include a reflection from the industry partner.

Seeding technologies: There has been a move in recent years towards the democratisation of decision making software development, driven by the increased popularity of open source and cloud technologies [10]. The increasing adoption of open source software has alleviated commercial software licensing costs, which were traditionally prohibitive to individuals. Similarly, the significant capital expenditure once associated with hardware purchasing has now been ameliorated by cloud computing. Consequently, software developer preference is gaining influence in organisations' technology procurement decisions.

Naturally, vendors now seek to target developers early in their career, exposing them to their technologies in the hope

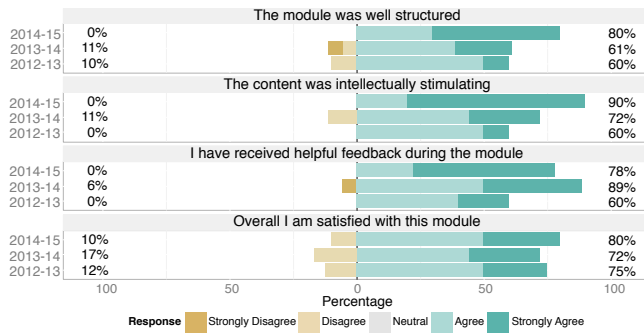


Figure 2: Student survey results

that they will go on to use these in their career. An effective mechanism which is increasingly appearing as part of many vendors' 'Developer Relations' strategies is to target future developers during their studies through collaboration on curriculum design, incorporating vendor technologies.

The industry partner is highly motivated to provide a positive experience for the students in using their tools and software. This encourages their continued use of the product upon completion of the course.

Talent acquisition: Our industry partner has an established talent acquisition pipeline for our students. Students may apply for internships while carrying out their capstone project, working closely with industry over a period of four months. The course has led to a number of internships being offered to students, some leading to job offers. Internship and employment statistics are presented in Section 7.

Work experience has been shown to be highly beneficial for graduate employment [4], and students' contributions to open source projects throughout their internship further enhances their appeal [10].

Our industry partner expresses positive sentiments around the initiative; *"In my opinion an academic initiative like this has the potential to offer a good return on investment whilst also being very low risk."*

Overall Reflection: *"The opportunity to have an input into the design and delivery of the CS curriculum provides Red Hat with access to graduates with skills valued by industry. Given the increasing skills gap and disparity between the delivery of education and the needs of employers, this relationship is critical for providing a continued pipeline of students into industry, and the development of a suitable curriculum which benefits the students."*

6. STUDENT SATISFACTION

Here we consider student satisfaction evaluation, using formal University surveys and volunteered testimonials.

The results from a University-administered survey, on a five-point Likert scale, are presented in Figure 2. We began a major coursework refresh to include cloud technologies at the beginning of the 2013/14 academic year, so we consider feedback following the update to better ensure our results are comparable.

Our results suggest that our ongoing efforts to enhance the course have lead to students considering the course better structured and more intellectually stimulating than previous years. We also observe a significant increase in the number of students who 'Strongly Agree' that they have re-

ceived helpful feedback. Despite increasing complexity of the coursework assignment in recent years, we see increases in overall satisfaction with the course.

Furthermore, we supplement these survey results with volunteered testimonials from our alumni, which suggest that students value the course's real-world and collaborative elements.

"This gave us an idea how challenging it is to tackle actual real world problems."

"covers a lot of aspects that are neglected or not thought about when doing assignments from other programming modules."

"helped students to collaborate and share ideas which each other when stuck at a problem."

"These are the type of situations that industries face and that require soft skills to manage and make decisions appropriately."

A number of students who have since graduated and joined industry discuss the course's benefits when seeking employment; e.g.: *"I believe this experience was highly valuable especially during the job hunting period for a student as I could mention about concepts and technologies learnt doing this assignment which are valuable to the industry."*

This positive feedback reinforces the practices and tools we have put in place on the course, highlighting areas of strength to further enhance in future iterations of the course.

7. STUDENT OUTCOMES

Table 7 shows the results of an investigation into the employment status of graduates, that have undertaken the *Enterprise Middleware* course over the last five years. Results were obtained through; the University alumni association, personal contact, and professional website LinkedIn. We also show the number of Red Hat internship, job and PhD studentship offers made to students from each cohort.

We observe a significant increase in cohort size from 2011-12 onwards. This is a consequence of the course being offered on a new MSc programme. Furthermore, 2013-14 sees a larger intake of international students, a trend which has continued in current academic year 2014-15. All graduates we have made contact with report they are either in relevant employment or undertaking further study.

We see that Red Hat consistently offers between one and two paid internships to students each year, which may lead to future employment. There is further evidence of Red Hat valuing doctoral study, providing a number of partially- and fully-funded PhD studentships.

Overall response rates for cohorts range between 55-87.5%. The Higher Education Statistics Agency (HESA) Destinations of Leavers from Higher Education (DLHE) survey³ seeks response rates of 75% for home students and 20% for international students, which we achieve for each cohort. Despite maintaining close contact with Home UK/EU students, with 77.8-100% response rate, we experienced lower response rates from our International graduates. Contributing factors to this are considered by [12].

We consider not only this snapshot of graduates' current employment status, but also explore the stability and quality of employment [17]. We further examine graduates' employment history, finding destinations to be stable and at an appropriate level (e.g. 'software developer', 'systems devel-

³<https://www.hesa.ac.uk/stats-dlhe>

Year	Program Enrollment			Survey Responses			Employed			Red Hat Offers		
	Total	Home	Int.	Total	Home	Int.	Industry	PhD	Other	Intern	Job	PhD
2013–14	27	7 (25.9%)	20 (74.1%)	16 (59.3%)	6 (85.7%)	10 (50%)	10	5	1	2	2	0
2012–13	14	7 (50%)	7 (50%)	9 (64.3%)	6 (85.7%)	3 (42.9%)	7	2	—	2	1	1
2011–12	20	9 (45%)	11 (55%)	11 (55%)	7 (77.8%)	4 (36.4%)	8	3	—	1	1	0
2010–11	7	3 (42.9%)	4 (57.1%)	5 (71.4%)	3 (100%)	2 (50%)	5	0	—	1	2	2
2009–10	8	5 (62.5%)	3 (37.5%)	7 (87.5%)	5 (100%)	2 (66.7%)	7	0	—	1	1	0

Table 1: Employment statistics for graduates over the last five years, including Red Hat job offer statistics.

oper’). When considering students from earlier cohorts, we see clear pathways to career progression, with many graduates entering senior and managerial roles.

Further work is required to conduct a longitudinal study to capture graduate career progression, alongside qualitative feedback on the impact of the course. In light of our ongoing interest in ensuring the applicability of our course to global job markets, particular attention will be given to maintaining better contact with international students.

8. OPEN SOURCE COURSEWARE

Based on the successes seen within Newcastle University, we have made the course materials available open source on GitHub⁴, for delivery at other institutions. These materials include a detailed tutorial introducing the tools used in the coursework and their configuration, and a detailed coursework specification. Model solutions and sample mark schemes are also available, stored in a private repository limiting access to instructors only.

This delivery of coursework materials also represents industry best practices for software documentation, written in AsciiDoc and made available through version control.

Delivering the content externally is mutually beneficial. From an academic perspective, the delivery of the course in other institutions leads to further scrutiny and accepted enhancements to the coursework materials will benefit the delivery of the course in subsequent years.

Furthermore, close cooperation between academia and industry for the production of courseware naturally requires significant time, effort, and resources. Thus, providing our materials for use externally is of huge benefit to institutions that may not have sufficient access to such resources.

Our industry partner comments; “*I see providing general material to multiple universities as a great way to scale the program, reaching a much greater number of students with only a slight increase in resources.*”

Our open source course content was first delivered remotely in a trial at Durham University, UK in the 2014-15 academic year to a cohort of 30 students. We now seek to engage with other institutions and industry to highlight the issues around delivering an open courseware with external institutions, by means of a cross-site study.

9. REFERENCES

- [1] Red Hat Inc. (Homepage). <http://redhat.com/>.
- [2] M. Ben-Ari. Constructivism in computer science education. *JCMST*, 20(1):45–73, 2001.
- [3] D. P. K. Chandran, Y. K. Sow, M. Harizan, C. C. Kooi, T. T. Hoy, and C. K. Foong. Success Story Of Collaboration Between Intel And Malaysian Universities To Establish And Enhance Teaching And Research In Electronic Packaging. In *IEEE IEMT*, 2010.
- [4] L. Harvey. Embedding and integrating employability. *New Directions for Institutional Research*, 2005(128):13–28, 2005.
- [5] M. J. Hawthorne and D. E. Perry. Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities. In *Software Engineering Education in the Modern Age*, pages 166–185. 2006.
- [6] C. K. Hobbs and H. H. Tsang. Industry in the Classroom: Equipping Students with Real-World Experience A reflection on the effects of industry partnered projects on computing education. In *ACM WCCCE ’14*, 2014.
- [7] P. G. Larsen, J. M. Fernandes, J. Habel, H. Lehrskov, R. J. Vos, O. Wallington, and J. Zidek. A multidisciplinary engineering summer school in an industrial setting. *EJEE*, 34(6):511–526, 2009.
- [8] T. C. Lethbridge. A survey of the relevance of computer science and software engineering education. In *IEEE CSEED ’98*, 1998.
- [9] D. Nicol. From monologue to dialogue: improving written feedback processes in mass higher education. *Assessment & Evaluation in Higher Education*, 35(5):501–517, 2010.
- [10] S. O’Grady. *The New Kingmakers*. O’Reilly Inc., 2013.
- [11] S. Palkar. Industry-academia collaboration, expectations, and experiences. *ACM Inroads*, 4(4):56–58, December 2013.
- [12] S. R. Porter and P. D. Umbach. Student survey response rates across institutions: Why do they vary? *Research in Higher education*, 47(2):229–247, 2006.
- [13] P. Race. *A briefing on self, peer & group assessment*, volume 9. LTSN, 2001.
- [14] A. Radermacher and G. Walia. Gaps between industry expectations and the abilities of graduates. In *SIGCSE ’13*, pages 525–530. ACM, 2013.
- [15] A. Rusu, R. Docimo, C. Santiago, and M. Paglione. Academia-academia-industry collaborations on software engineering projects using local-remote teams. In *SIGCSE ’09*, pages 301–305. ACM, 2009.
- [16] J. Tenenbergs. Industry fellows: bringing professional practice into the classroom. In *ACM SIGCSE*, 2010.
- [17] A. Tymon. The student perspective on employability. *Studies in higher education*, 38(6):841–856, 2013.
- [18] G. Ushaw, W. Blewitt, and G. Morgan. Adopting commercially inspired practices within an academic teaching course: A case study of a computer games engineering degree. In *CSEDU ’14*, 2014.

⁴<http://EnterpriseMiddlewareCoursework.github.io/>